

AD-A036 713

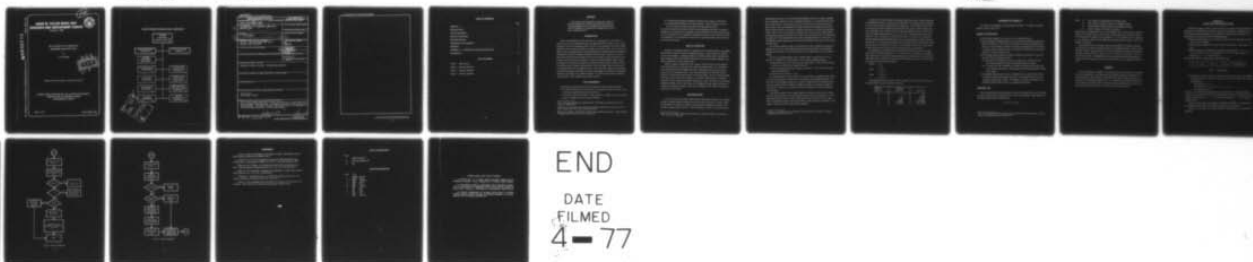
DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/6 9/2  
THE FEASIBILITY OF A METHOD OF PROCESSING ENCRYPTED DATA.(U)  
JAN 77 L M CULPEPPER

UNCLASSIFIED

CMLD-77-02

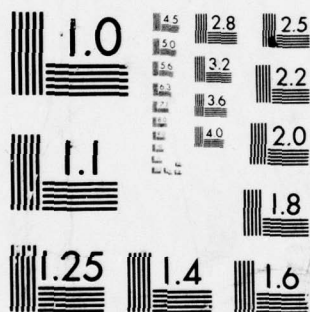
NL

| OF |  
AD  
A036 713



END

DATE  
FILMED  
4-77



MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

ADA036713

THE FEASIBILITY OF A METHOD OF PROCESSING ENCRYPTED DATA

Report CMLD-77-02

**DAVID W. TAYLOR NAVAL SHIP  
RESEARCH AND DEVELOPMENT CENTER**

Bethesda, Md. 20084



**THE FEASIBILITY OF A METHOD OF  
PROCESSING ENCRYPTED DATA**

by

L. M. Culpepper



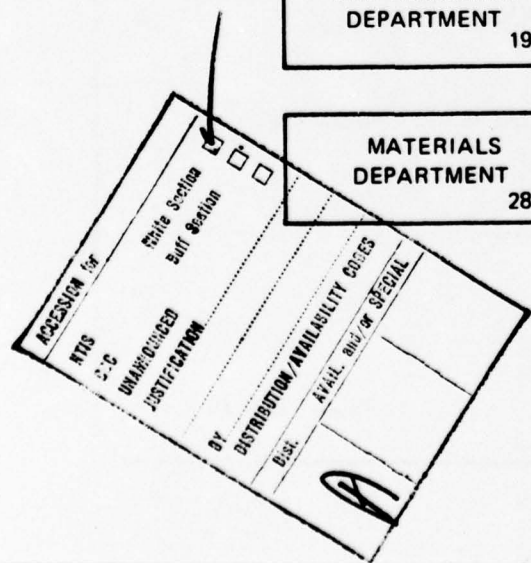
APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

s/c  
3 COMPUTATION, MATHEMATICS, AND LOGISTICS DEPARTMENT  
RESEARCH AND DEVELOPMENT REPORT  
DEPARTMENTAL REPORT

January 1977

Report CMLD-77-02

## MAJOR DTNSRDC ORGANIZATIONAL COMPONENTS



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER DTNSRDC Report <u>CMLD-77-02</u>	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) <u>THE FEASIBILITY OF A METHOD OF PROCESSING ENCRYPTED DATA.</u>	5. TYPE OF REPORT & PERIOD COVERED	
6. AUTHOR(s) <u>L. M. Culpepper</u>	7. CONTRACT OR GRANT NUMBER(s)	
8. PERFORMING ORGANIZATION NAME AND ADDRESS <u>David W. Taylor Naval Ship R&amp;D Center Bethesda, Maryland 20084</u>	9. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS <u>O&amp;MN 1-1830-002</u>	
10. CONTROLLING OFFICE NAME AND ADDRESS	11. REPORT DATE <u>11 Jan 1977</u>	12. NUMBER OF PAGES <u>12 16 p.</u>
13. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	14. SECURITY CLASS. (of this report) <u>Unclassified</u>	
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE		
16. DISTRIBUTION STATEMENT (of this Report)  APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)  ADP Security Intelligent Terminal		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)  The time and expense required to certify the security of a multi-level multi-access computer frequently make it necessary to process classified data in a dedicated mode. This report discusses the feasibility of a limited type of classified data processing in a multi-access mode.		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE  
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)



## TABLE OF CONTENTS

	Page
ABSTRACT .....	1
INTRODUCTION .....	1
THE ENVIRONMENT .....	1
MODE OF OPERATION .....	2
IMPLEMENTATION .....	2
DISCUSSION OF FEASIBILITY .....	5
SUMMARY .....	6
APPENDIX A – B-TREES AND THEIR MANIPULATION .....	7
REFERENCES .....	12

## LIST OF FIGURES

Figure 1 – Page Structure .....	7
Figure 2 – Retrieval Algorithm .....	9
Figure 3 – Insertion Algorithm .....	10
Figure 4 – Deletion Algorithm .....	11

## ABSTRACT

The time and expense required to certify the security of a multi-level multi-access computer frequently make it necessary to process classified data in a dedicated mode. This report discusses the feasibility of a limited type of classified data processing in a multi-access mode.

## INTRODUCTION

Privacy transformations have been proposed as a countermeasure to several types of threats to the security of information stored in a large, multi-access computer system. Turn<sup>1</sup> summarizes some of the problems associated with the use of transformed data bases including distribution of keys, selective updating, and change of key, among others. Turn's view of the utility of privacy transformations for data banks seems to be oriented towards the problem of protecting the contents of physically removable data storage units. Feistel et al.<sup>2</sup> have suggested that on-line direct-access storage devices can be protected if file-access methods can be devised to deal directly with cryptograms. No results have been published however. Bayer and Metzger<sup>3</sup> have studied the problem of protecting information stored in indexed, random access files by means of privacy transformations. They also consider threats to the security of a transformed file due to updating and to the file structure itself. Based on these findings, an approach is proposed which shifts the primary responsibility for protecting the information from the central facility to an intelligent terminal. The software required for the intelligent terminal could be made certifiable using available methods.

## THE ENVIRONMENT

The information processing system under consideration has the following elements:

- A central processing facility under control of a commercially available operating system and having the appropriate mass storage hierarchy.
- A collection of files of information – some of which are available for general use while others are available only with proper authorization.

---

<sup>1</sup>Turn, R., "Privacy Transformations for Databank Systems," Proceedings of the National Computer Conference, pp. 589-601 (1973).

<sup>2</sup>Feistel, H. et al., "Some Cryptographic Techniques for Machine-to-Machine Data Communication," Proceedings of the IEEE, Vol. 63, No. 11, pp. 1545-1554 (Nov 1975).

<sup>3</sup>Bayer, R. and J. Metzger, "On Encipherment of Search Trees and Random Access Files," ACM Transactions on Database Systems, Vol. 1, No. 1, pp. 37-52 (Mar 1976).

- A communications network consisting of common carrier, leased, or private lines.
- A geographically distributed set of query terminals – some of which are in areas to which the public has access while others are in areas which are secure.

The system considered in this report is used for query/transaction processing. Transactions originate at the terminals and are transmitted over the network to the central processing facility. The appropriate files are accessed, the transaction is processed, and the transaction result is transmitted over the communications network to the requesting terminal where additional processing may take place.

## MODE OF OPERATION

Protection of information stored in a system such as that described above is an extremely difficult problem which has received considerable attention. Entirely satisfactory solutions have not been obtained. A practical approach to the problem which utilizes currently available technology, affords a high degree of protection for a somewhat limited application, and which is reasonable in cost, is outlined below.

Consider a group of analysts who in the course of their work require conversational access to several files, one of them large ( $10^7$  entries) and highly sensitive. The analysts perform retrievals, deletions, and updates on the files. Other groups of analysts may require access to only the non-sensitive files. In the proposed approach, all of the files would be stored in the on-line direct access memory of the central facility. The file containing sensitive information would be protected by storing it in a suitably enciphered form. Transactions (retrievals, deletions, updates) involving the sensitive file would be performed by analysts using intelligent terminals located in secure areas. The intelligent terminals would perform the deciphering and enciphering necessary to process the transactions. The details of the approach are given in the next section.

## IMPLEMENTATION

In order to provide a reasonable response time for transactions processed at the intelligent terminal, a suitable method for organizing the file for remote access must be selected. Bayer and McCreight<sup>4</sup> have studied the problem of organizing and maintaining an index for a dynamically changing random access file. In their terminology, an index is a collection of index

---

<sup>4</sup>Bayer, R. and E. McCreight, "Organization and Maintenance of Large Ordered Indices," Acta Informatica, Vol. 1, No. 3, pp. 173-189 (1972).

elements which are pairs  $(x, a)$  of data items stored together, where  $x$  is a name or identifier and  $a$  is (usually) a pointer into a file of associated information. It is assumed that the index may be quite large, say  $10^7$  elements, and that it is stored on direct access devices such as discs or drums. Bayer and McCreight have devised a method for organizing the index which allows insertion, retrieval, and deletion of elements in the index in an amount of time proportional to  $\log_k N$  where  $N$  is the number of elements in the index and  $k$  is a natural number describing the page size most suitable for the direct access device being used. It is shown that for a page size of 120 index elements and an index containing between  $4.5 \times 10^5$  and  $2.1 \times 10^8$  elements, an index element can be retrieved from the random access index file in no more than four reads.

In the scheme described by Bayer and McCreight, the index is organized into pages containing between  $k$  and  $2k$  index elements which are stored sequentially in increasing order. The pages are the nodes of a recently defined type of tree called a B-tree which is defined as follows:

Let  $h$  be a non-negative integer, and  $k$  a positive integer. A directed tree  $T$  is in the class  $t(k, h)$  of B-trees if  $T$  is either empty ( $h = 0$ ) or has the following properties:

- Each path from the root to any leaf has the same length  $h$ , which is called the height of the tree;  $h$  is the number of nodes in the path.
- Each node except the root and the leaves has at least  $k + 1$  sons. The root is a leaf or has at least two sons.
- Each node has at most  $2k + 1$  sons.

It is proposed that the index be stored in the central facility in enciphered form in pages of size  $2k$  and accessed by a standard direct access method such as BDAM in OS/360. The role of the remote intelligent terminal is to accept the transaction from the analyst and perform the necessary retrieval, deciphering, enciphering, insertion, and deletion operations on the index file. The direct access method on the central facility is used only to manipulate encrypted pages.

As noted by Anderson,<sup>5</sup> the technique selected to encipher the index file is of vital importance since the structure of the file provides valuable clues to the cryptanalyst. It is proposed that the file be enciphered with a block cipher using a different key for each page. Bayer and Metzger<sup>3</sup> describe a scheme which seems to satisfy Anderson's requirements. Their scheme is summarized briefly in the following paragraph.

---

<sup>5</sup> Anderson, J., "Information Security in a Multi-User Computer Environment," *Advances in Computers*, M. Rubino, Editor, New York (1972).

Consider a file  $F$  which is stored on secondary memory in  $m$  pages. Let each page have an associated page number  $p$  to locate it and a page ID  $\bar{p}$ . The plain and cipher text versions of the  $i^{\text{th}}$  page are denoted by  $Q(p_i)$  and  $C(p_i)$  respectively. Two ciphers are used for file encrypting, a text cipher  $U$  (which must be reversible) and a page key cipher  $E$  (which need not be reversible). If  $K_E$  is the key for  $E$ , then, given the page ID  $\bar{p}_i$  for  $p_i$ ,  $E$  is used to calculate the corresponding page key  $k_{p_i}$  by  $k_{p_i} = E(\bar{p}_i, K_E)$ . The page contents are then encrypted by  $C(p_i) = U(Q(p_i), k_{p_i})$  and decrypted by  $Q(p_i) = U^{-1}(C(p_i), k_{p_i})$ . Bayer and Metzger suggested assigning arbitrary values to the  $\bar{p}_i$  and storing them in a table, ordered by page number. In our proposal, the page ID could be generated from the page number using a suitable cipher and a separate key if desired. In this approach, each user of a file would be required to know two keys,  $K$  and the key required to generate the page ID's.

The algorithms for retrieval, insertion, and deletion of nodes on the B-tree are rather simple and are discussed in Appendix A. If  $h$  is the height of a given B-tree,  $f$  the number of pages which must be fetched from secondary storage for a transaction, and  $w$  the number of pages which must be written, then the number of fetches/writes required for the three operations in the worst case are given as follows:

Retrieve	$f = h$ $w = 0$
Delete	$f = 2h - 1$ $w = h + 1$
Insert	$f = 3h - 1$ $w = 2h + 1$

The following gives the size range, in terms of elements, of index files which may exist for various values of  $h$  and a page size of 120 items.

Height of Page Tree	Minimum Index Size	Index Size
1	1	120
2	121	14640
3	7441	1771560
4	453961	214358880
5	27691681	$2.59 \times 10^{10}$

## DISCUSSION OF FEASIBILITY

Two aspects of the feasibility of this approach are discussed – the degree of protection afforded, and the response time.

### DEGREE OF PROTECTION

- The intelligent terminal can be placed in a secure, shielded enclosure.
- Active and passive wire-tapping threats can be countered using techniques described in Kent<sup>6</sup> and Feistel et al.<sup>2</sup> Kent presents the design of protocols and protection modules for a host and intelligent terminals. Kent claims his design prevents the disclosure of message contents, provides detection of message stream modification, and provides detection of denial of message service.
- The encryption scheme described by Bayer and Metzger was designed for structured files and includes the following features:
  - (a) Each block of information in the file is encrypted with a different key thus countering cryptanalysis techniques requiring long cipher text strings.
  - (b) If a block of the file is deciphered, it provides the cryptanalyst with little assistance in breaking other blocks. Defenses against this threat are discussed at length in their paper.
- The approach described does not rely on certification of the central system software to provide data protection although some degree of certification may be required to provide the necessary data integrity. The approach does require certification of the intelligent terminal software which is a comparatively simple task.

### RESPONSE TIME

Bayer and McCreight<sup>4</sup> provide estimates for the number of pages from an index file which must be examined to perform a simple retrieval. Based on their example, timing estimates for the scheme proposed here are determined by

$$T = (T_a + T_t + T_d)h$$

---

<sup>6</sup>Kent, S., "Encryption-Based Protection Products for Interactive User-Computer Communication," MIT Laboratory for Computer Science Report 162 (May 1976).

where  $T_a$  Time required to access a page on the central facility  
 $T_t$  Time required to transmit a page to the intelligent terminal  
 $T_d$  Time required to decipher the page in the intelligent terminal  
 $h$  Number of pages to be fetched and searched.

Assume that the processing time for the retrieval algorithm itself may be neglected. Assume a page size of 120 index elements, each 14 bytes. With this page size, an index file of up to  $2 \times 10^8$  entries can be stored in a tree of height  $h = 4$ . Let us assume a value of 100 ms for  $T_a$ . To calculate  $T_t$ , assume we have 4800 bps communication lines. Thus  $T_t = 120 \times 14 \times 8 / 4800 = 2.8$  seconds.

Assume that the file was encrypted using the National Bureau of Standards algorithm, and that 50 ms are required to decipher 8 bytes on the intelligent terminal. Assume also that each page is scanned using a binary search technique and that index elements are deciphered only when necessary. Then  $T_d = (\log_2 120 - 1) 14 \times 0.05 / 8 = 0.5$  and thus  $T = (0.1 + 2.8 + 0.5)4 = 14.6$  seconds. If deciphering were done in hardware and with wider bandwidth communications, the response time could be greatly reduced, even if a progressive cipher were used.

## SUMMARY

This report discusses the feasibility of performing operations on data stored in encrypted form on a central facility. It has shown that using available techniques, transactions on encrypted files may be performed with a reasonable response time and with a high degree of protection on commercially available systems. It is envisioned that intelligent terminals used for this purpose in the future will be more powerful personal computers, and that the role of the central facility will become that of a back-end data management system.

## APPENDIX A

### B-TREES AND THEIR MANIPULATION

Recall that the pages on which the index is stored are the nodes of a B-tree and that they contain up to  $2k$  elements where an element consists of an identifier ( $x$ ), a pointer ( $a$ ) to a file containing associated information, and a pointer ( $p$ ) to another page in the index. The data structure of the index also has the following properties:

- Each page holds between  $k$  and  $2k$  elements except for the root page which may hold between 1 and  $2k$  elements.
- Let the number of elements on a page  $P$ , which is not a leaf, be  $L$ . Then  $P$  has  $L + 1$  sons.
- Within each page  $P$ , the elements are sequential in increasing order:  $x_1, x_2, \dots, x_L$ ;  $k \leq L \leq 2k$  except for the root page for which  $1 \leq L \leq 2k$ .  $P$  also contains  $L + 1$  pointers  $p_0, p_1, \dots, p_L$  to the sons of  $P$ .

The logical structure of a page is shown in the following figure.

$p_0$	$x_1 a_1 p_1$	$x_2 a_2 p_2$	$\dots$	$x_L a_L p_L$	(Unused Space)
-------	---------------	---------------	---------	---------------	----------------

Figure 1 – Page Structure

- Let  $P(p_1)$  be the page to which  $p_1$  points, let  $k(p_1)$  be the set of keys on the pages of that maximal subtree of which  $P(p_1)$  is the root. Then for these B-trees the following conditions hold:

$$[\forall y \in k(p_0)] (y < x_1)$$

$$[\forall y \in k(p_i)] (x_i < y < x_{i+1}) \quad i = 1, 2, \dots, L - 1$$

$$[\forall y \in k(p_L)] (x_L < y)$$

Now, let  $p, r, s$  be pointer variables which may assume the value "U" meaning undefined,  $r$  points to the root and is U if the tree is empty, and let  $y$  be the key. Let  $P(p)$  be the page to which  $p$  is pointing, then  $x_1, \dots, x_L$  are the keys in  $P(p)$  and  $p_0, \dots, p_L$  are the page pointers in  $P(p)$ .

Figure 2 is a flow chart of the algorithm for retrieving an identifier. In actual practice, the sequential searches within a node would be replaced by a binary search.

Figure 3 is a flow chart of the insertion algorithm. The split page routine performs the following:

- Insert the entry into the sequence of entries in P (in main store) resulting in  $p_0, (x_1, p_1), (x_2, p_2), \dots, (x_{2k+1}, p_{2k+1})$ .
- Put the subsequent  $p_0, (x_1, p_1), \dots, (x_k, p_k)$  into P and introduce a new page P' to contain the subsequent  $p_{k+1}, (x_{k+2}, p_{k+2}), \dots, (x_{2k+1}, p_{2k+1})$ . Let Q be the father page of P. Insert the entry  $(x_{k+1}, p')$  where  $p'$  points to P' into Q. P' then becomes a brother of P. These pages are appropriately encrypted and transmitted to the central facility.

The deletion algorithm is shown in Figure 4.

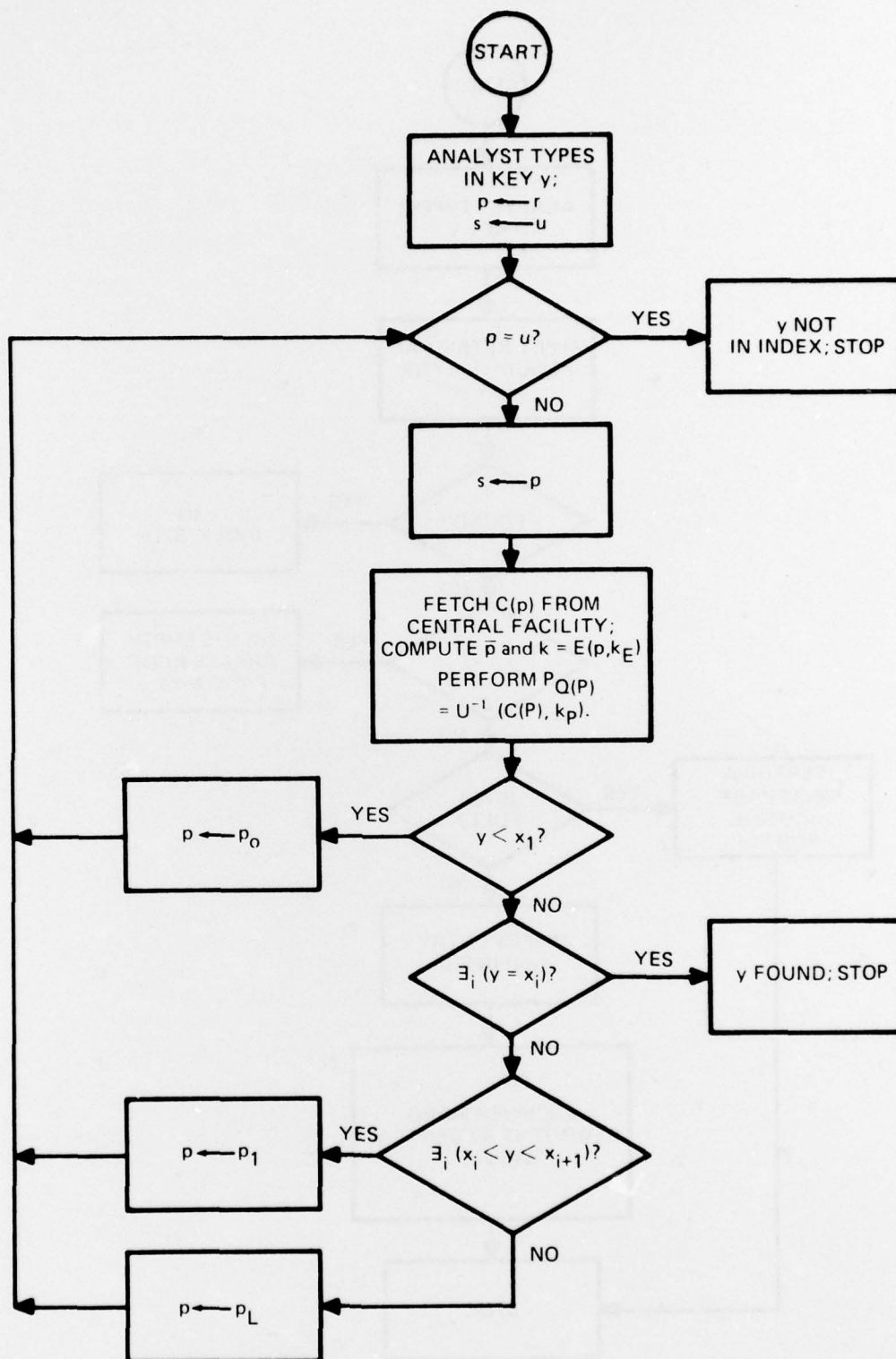


Figure 2 - Retrieval Algorithm

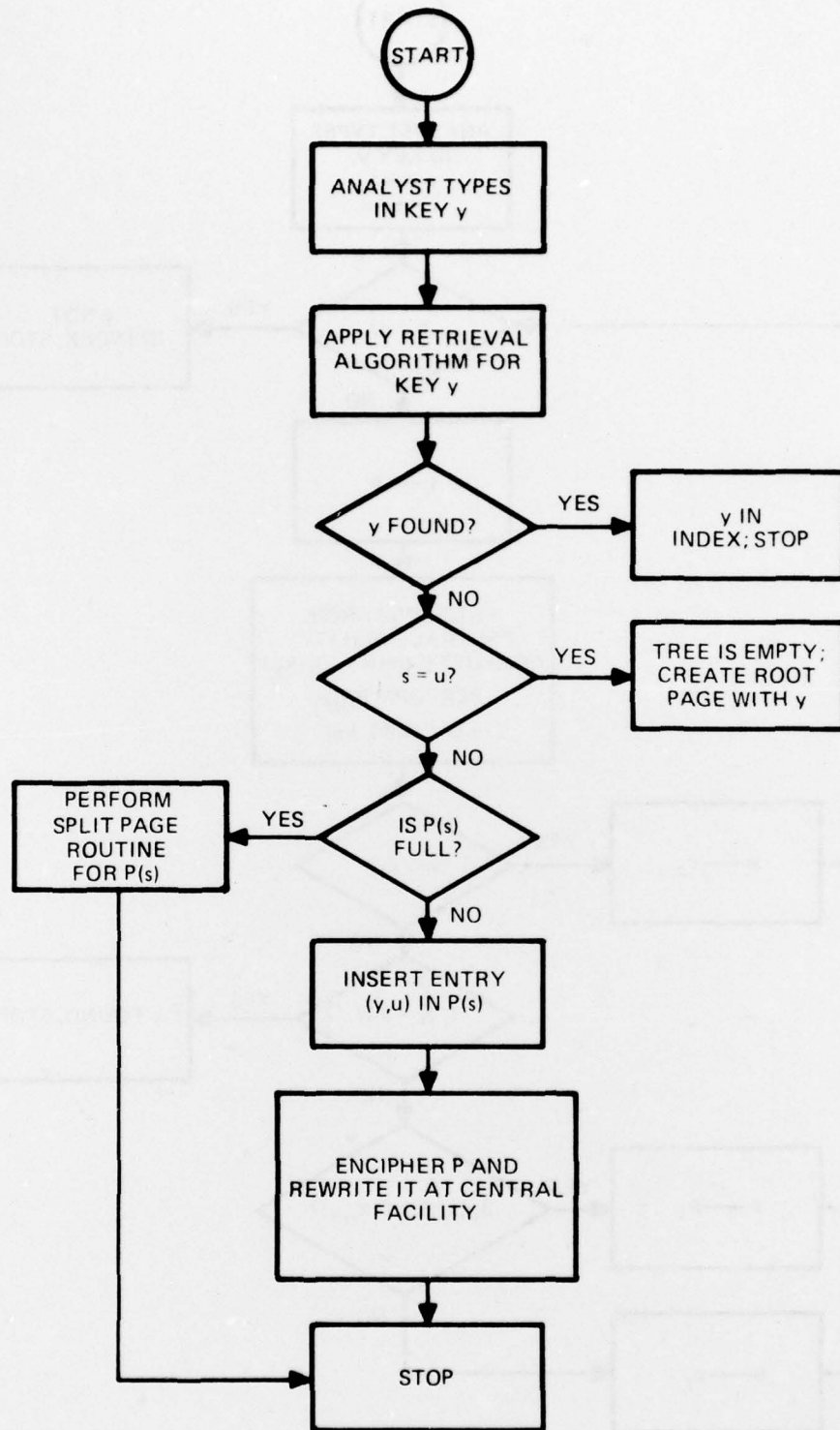


Figure 3 - Insertion Algorithm

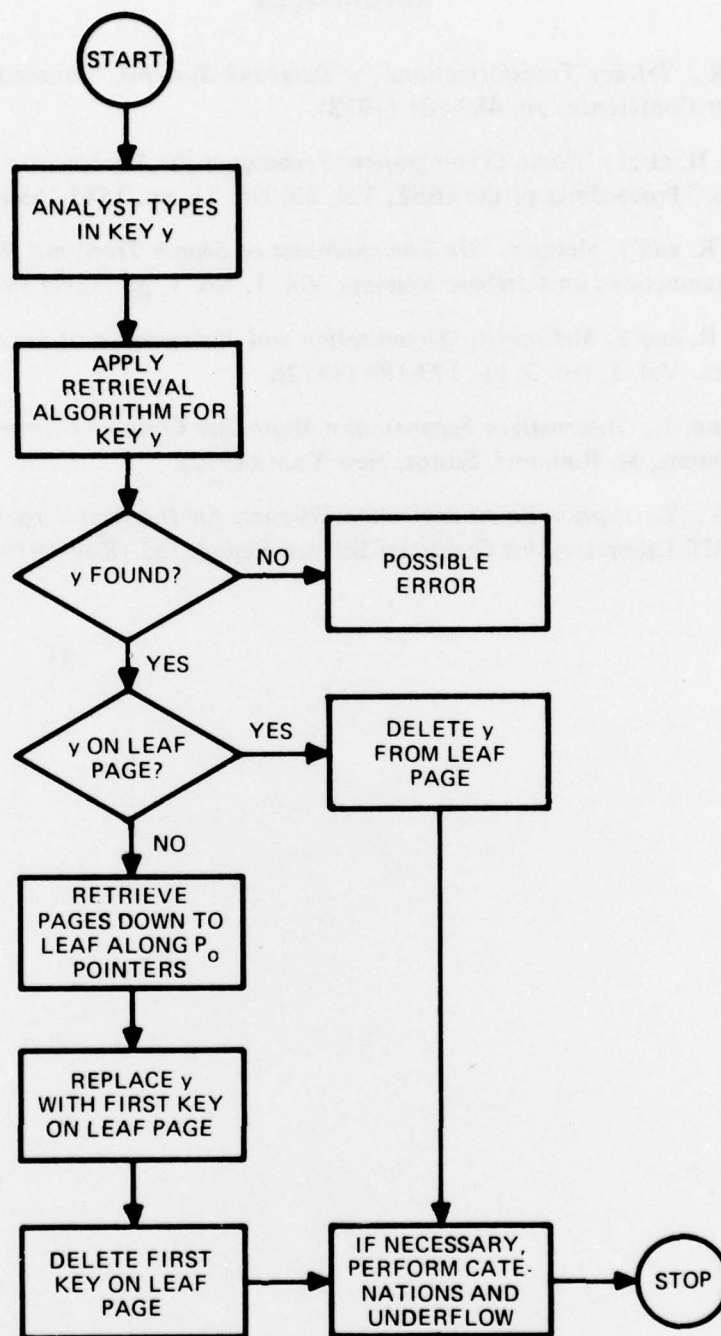


Figure 4 - Deletion Algorithm

## REFERENCES

1. Turn, R., "*Privacy Transformations for Databank Systems*," Proceedings of the National Computer Conference, pp. 489-601 (1973).
2. Feistel, H. et al., "*Some Cryptographic Techniques for Machine-to-Machine Data Communication*," Proceedings of the IEEE, Vol. 63, No. 11, pp. 1545-1554 (Nov 1975).
3. Bayer, R. and J. Metzger, "*On Encipherment of Search Trees and Random Access Files*," ACM Transactions on Database Systems, Vol. 1, No. 1, pp. 37-52 (Mar 1976).
4. Bayer, R. and E. McCreight, "*Organization and Maintenance of Large Ordered Indices*," Acta Informatica, Vol. 1, No. 3, pp. 173-189 (1972).
5. Anderson, J., "*Information Security in a Multi-User Computer Environment*," Advances in Computers, M. Rubinoff, Editor, New York (1972).
6. Kent, S., "*Encryption-Based Protection Products for Interactive User-Computer Communication*," MIT Laboratory for Computer Science Report 162 (May 1976).

## INITIAL DISTRIBUTION

### Copies

1	CHONR 437/Denicoff
10	NAVELEX 5703/Worthman
12	DDC

## CENTER DISTRIBUTION

### Copies

### Code

1	18/1808	Gleissner
1	1802.2	Frenkiel
1	1802.4	Theilheimer
1	1805	Cuthill
2	1809	Harris
10	1835	Culpepper
1	184	Lugt
1	185	Corin
1	186	Sulit
1	189	Gray
1	1892.1	Strickland

**DTNSRDC ISSUES THREE TYPES OF REPORTS**

(1) DTNSRDC REPORTS, A FORMAL SERIES PUBLISHING INFORMATION OF PERMANENT TECHNICAL VALUE, DESIGNATED BY A SERIAL REPORT NUMBER.

(2) DEPARTMENTAL REPORTS, A SEMIFORMAL SERIES, RECORDING INFORMATION OF A PRELIMINARY OR TEMPORARY NATURE, OR OF LIMITED INTEREST OR SIGNIFICANCE, CARRYING A DEPARTMENTAL ALPHANUMERIC IDENTIFICATION.

(3) TECHNICAL MEMORANDA, AN INFORMAL SERIES, USUALLY INTERNAL WORKING PAPERS OR DIRECT REPORTS TO SPONSORS, NUMBERED AS TM SERIES REPORTS; NOT FOR GENERAL DISTRIBUTION.